Indian Institute of Technology Bombay

IITB CPU
Project Report

by

TEAM ID: 11
Takshit Aggarwal (22B1236)
Harsh Anand (22B1249)
Medhansh Sharma (22B1287)
Sarthak Niranjan (22B3923)

Course Instructor

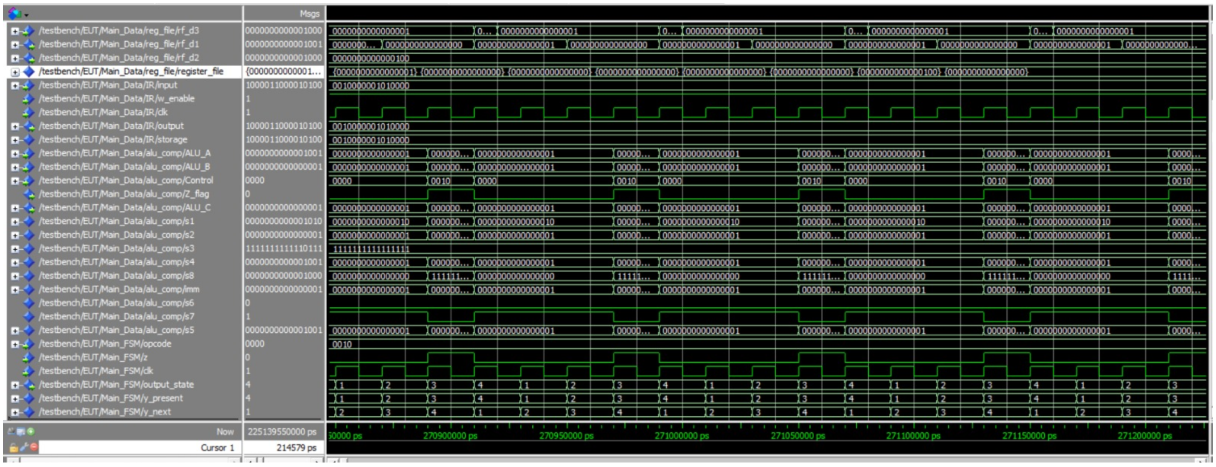
Prof. Virendra Singh,
Department of Electrical Engineering,
IIT Bombay.

EE 224 : Digital Systems
Course Project
Autumn, 2023

Objective

This project required us to develop a 16-bit computing system named IITB-CPU with a specified instruction set architecture. We were required to implement the system using VHDL as the Hardware Description Language (HDL). The IITB-CPU should features an 8-register architecture and operate on 16-bit data, utilizing a point-to-point communication infrastructure for efficient data exchange.

Simulation



```
                                                    Msgs
/testbench/EUT/Main_Data/reg_file/rf_d3             0000000000001000
/testbench/EUT/Main_Data/reg_file/rf_d1             0000000000001001
/testbench/EUT/Main_Data/reg_file/rf_d2             0000000000001000
/testbench/EUT/Main_Data/reg_file/register_file     {000000000001...
/testbench/EUT/Main_Data/IR/input                   1000011000010100
/testbench/EUT/Main_Data/IR/w_enable                1
/testbench/EUT/Main_Data/IR/clk                     1
/testbench/EUT/Main_Data/IR/output                  1000011000010100
/testbench/EUT/Main_Data/IR/storage                 1000011000010100
/testbench/EUT/Main_Data/alu_comp/ALU_A             0000000000001001
/testbench/EUT/Main_Data/alu_comp/ALU_B             0000000000000001
/testbench/EUT/Main_Data/alu_comp/Control           0000
/testbench/EUT/Main_Data/alu_comp/Z_flag            0
/testbench/EUT/Main_Data/alu_comp/ALU_C             0000000000000001
/testbench/EUT/Main_Data/alu_comp/s1                0000000000001010
/testbench/EUT/Main_Data/alu_comp/s2                0000000000000001
/testbench/EUT/Main_Data/alu_comp/s3                1111111111110111
/testbench/EUT/Main_Data/alu_comp/s4                0000000000001001
/testbench/EUT/Main_Data/alu_comp/s8                0000000000001000
/testbench/EUT/Main_Data/alu_comp/imm               0000000000000001
/testbench/EUT/Main_Data/alu_comp/s6                0
/testbench/EUT/Main_Data/alu_comp/s7                1
/testbench/EUT/Main_Data/alu_comp/s5                0000000000001001
/testbench/EUT/Main_FSM/opcode                      0000
/testbench/EUT/Main_FSM/z                           0
/testbench/EUT/Main_FSM/clk                         1
/testbench/EUT/Main_FSM/output_state                4
/testbench/EUT/Main_FSM/y_present                   4
/testbench/EUT/Main_FSM/y_next                      1
                                              Now   225139550000 ps
                                         Cursor 1   214579 ps
```

All States

S1

- RF_A1 -> 111
- Mem_Add -> RF_D1
- IR _in> Memory
- OpCode -> IR_out(15:12)

S2

- RF_A1 -> IR_out(11:0)
- RF_A2 -> IR_out(8:6)
- T1_in -> RF_D1
- T2 _in-> RF_D2
- RF_A3 -> IR_out(5:3)

S3

- OpCode -> IR(15:12)
- ALU_A -> T1_out
- ALU_B -> T2_out
- T3_in -> ALU_C

S4

- RF_A1 -> 111
- ALU_A -> RF_D1
- ALU_B -> 0000000000000001
- RF_A3 -> 111
- RF_D3 -> ALU_C

S5

- RF_A1 -> IR_out(11:9)
- T1_in -> RF_D1
- SE10_in -> IR_out(5:0)
- T2 -> SE10
- RF_A3 -> IR_out(8:6)

S6

- SE7_in -> IR_out(8:0)
- T1_in -> SE7_out
- RF_A3 -> IR_out(11:9)
- RF_D3 -> T1_out

S7

- SE7_in -> IR_out(8:0)
- L8_in -> SE7_out
- T1_in -> L8_out
- RF_A3 -> IR_out(11:9)
- RF_D3 -> T1_out

S8

- RF_A1 ->IR_out(11:9)
- RF_A2 -> IR_out(8:6)
- T1_in -> RF_D1
- T2_in ->RF_D2

S9

- RF_A1-> "111"
- ALU_A -> T1_out
- ALU_B ->T2_out
- ALU_A ->RF_D1
- SE10_in ->IR_out (5:0)
- ALU_b ->SE_10_out
- RF_A3 ->"111"
- RF_D3 ->ALU_C

S10

- RF_A1-> "111"
- ALU_A ->T1_out
- ALU_B -> T2_out
- ALU_A -> RF_D1
- ALU_b -> "0000000000000001"
- RF_A3 -> "111"
- RF_D3 ->ALU_C

S11

- RF_A1 ->"111"
- RF_A3 -> IR_OUT (11:9)
- RF_D3 ->RF_D1
- T1_in -> Rf_D1
- SE10_in ->IR_out (5:0)
- T2_in -> SE10_out

S12

- ALU_C ->T1_out
- AlLU_B ->T2_out
- RF_A3 ->"111"
- Rf_D3 ->ALU_C

S13

- RF_A1 -> "111"
- RF_A3 -> IR_out(11:9)
- RF_D3 ->Rf_D1

S14

- RF_A1 -> IR_out (8:6)
- RFRf__A3 -> "111"
- RF_D3 -> RF_D1

S15

- RF_A2 ->IR_out(8:6)
- ALU_B ->RF_D2
- SE10_in ->IR_out(5:0)
- ALU_A ->SE10_out
- M_A ->ALU_C

S16

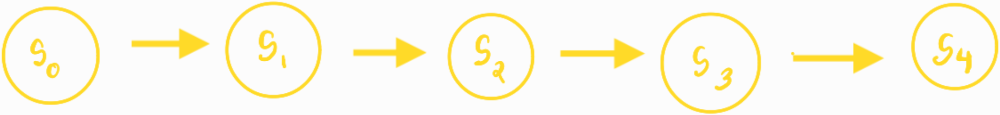
- RF_A3 -> IR_out (11:9)
- RF_D3 -> M_out

S17

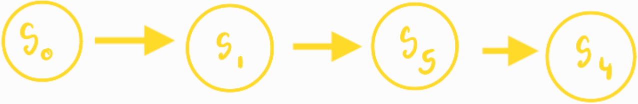
- RF_A1 ->IR_out (11:9)
- T1_in ->RF_D1
- M_in ->T1_out

Instruction State Diagrams

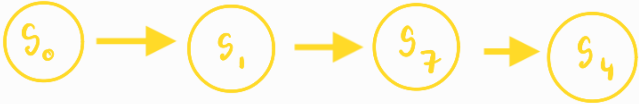
ADD / SUB / MUL / AND / ORA / IMP



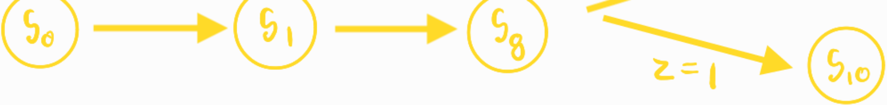
ADI



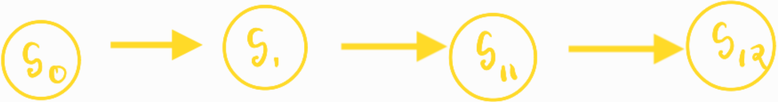
LHI/LLI



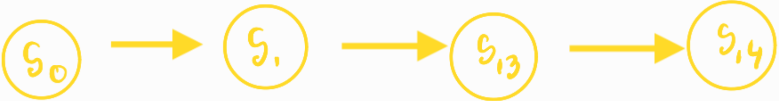
BEQ


S_0
S_1
S_8
$z = 1$
S_{10}

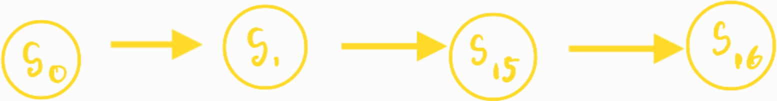
JAL


S_0
S_1
S_{11}
S_{12}

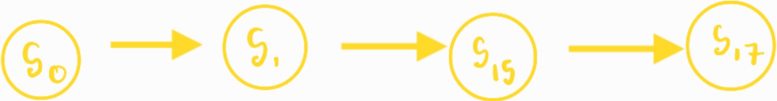
JLR

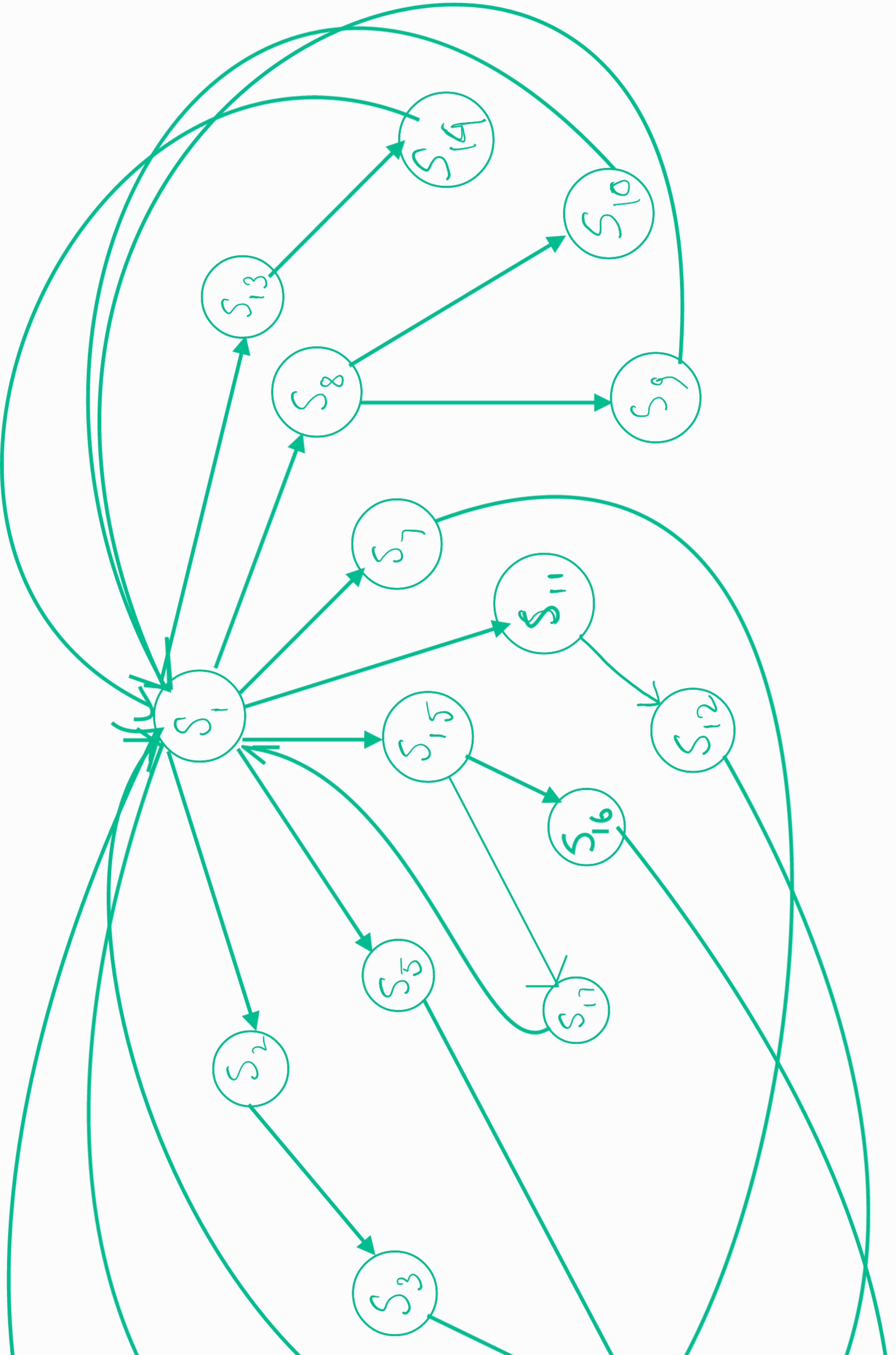

S_0
S_1
S_{13}
S_{14}

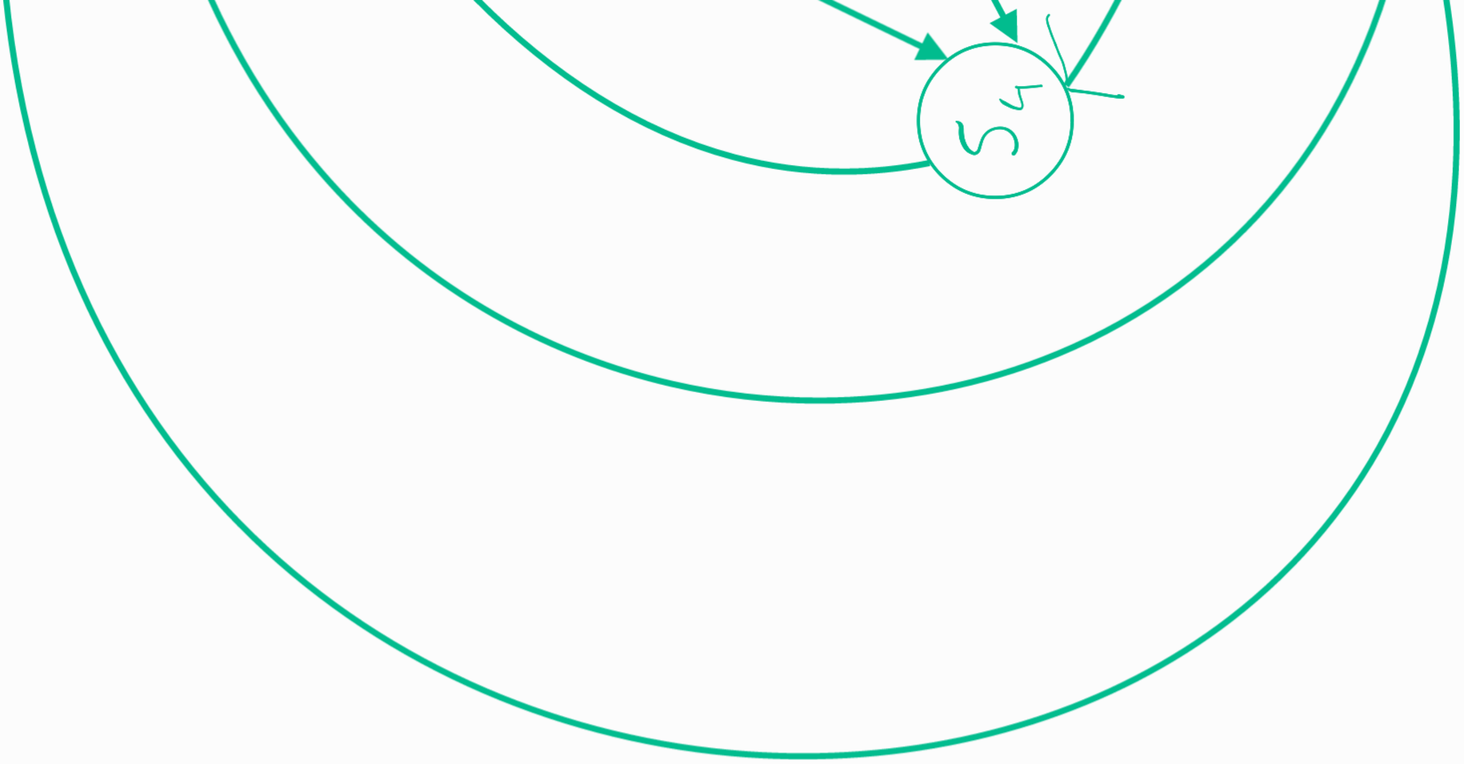
LW


S_0
S_1
S_{15}
S_{16}

SW


S_0
S_1
S_{15}
S_{17}


S14
S10
S13
S8
S9
S7
S11
S1
S15
S12
S16
S5
S17
S2
S3


S4

Contributions

Takshit Aggarwal (Project Leader) :

- Handled distribution and supervision of work among team members.
- Encoded the multiplexers used for controlling the Datapath in VHDL.
- Wrote the VHDL code for a hard-coded Memory.
- Helped in designing and coding the Datapath and Control signals for the system.
- Helped in Debugging the code to ensure proper functioning.

Harsh Anand :

- Designed the pen-paper State Machine Diagrams of the instructions.
- Encoded the registers used to store data in VHDL.
- Helped in encoding the state machines in VHDL.
- Contributed to compiling our progress in a detailed Project Report.

Medhansh Sharma :

- Encoded the components such as sign extender, register file and left shifters.
- Helped in designing and coding the Datapath and Control signals for the system.
- Wrote the VHDL code for ALU used for performing multiple different operations as required.
- Helped in Debugging the code to ensure proper functioning.

Sarthak Niranjan :

- Helped in designing and encoding the state machines in VHDL.
- Wrote the VHDL code for CPU used for integrating the Datapath with the State Machine encoded previously.
- Contributed to compiling our progress in a detailed Project Report.

Changes

January 30, 2024

- A 6x1 mux was added to the ALU to synchronise it with the clock of CPU
- ALU was coded in behavioral modelling earlier but as structural modelling gave better result so we changed the ALU to structural modelling
- A functional reset button was added to the CPU
- The LEDs were mapped to various registers and memory and the switches were used to alternate between them
- Some of the instructions whoser simulations were not working earlier were improved by adding extra states which made the other functions functional.
- A functional reset button was added to the CPU